PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE HONORABLE BOARD OF PATENT APPEALS AND INTERFERENCES

In re the Application of

Michael Weiss, et al.

Application No.: 09/768,129 Examiner: E.G. Milef

Filed: January 23, 2001 Docket No.: PERY 2 00002

For: CACHING MECHANISM TO OPTIMIZE A BIDDING PROCESS USED TO SELECT RESOURCES AND SERVICES

APPEAL BRIEF UNDER 37 C.F.R. 41.37

Appeal from Group 3628
FAY SHARPE LLP
1100 Superior Avenue – Seventh Floor
Cleveland, Ohio 44114-2579
Telephone: 216-861-5582
Attorneys for Appellants

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest for this appeal and the present application is Mitel Networks Corporation, by way of an Assignment recorded in the U.S. Patent and Trademark Office at Reel 016345, Frame 0283.

II. STATEMENT OF RELATED APPEALS AND INTERFERENCES

There are no prior or pending appeals, interferences or judicial proceedings, known to Appellants, Appellants' representative, or the Assignee, that may be related to, or which will directly affect or be directly affected by or have a bearing upon the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

All pending claims (1-17) were finally rejected in the Office Action dated March 21, 2007.

Appellants appeal the rejection of the pending claims, namely, claims 1-17.

IV. STATUS OF AMENDMENTS

An Amendment After Final was filed August 16, 2007. The proposed amendment to Claim 1 was as follows:

> A storage medium storing a set of program instructions executable on a data processing device and usable to create agents and adapters for optimizinge a bidding process for resources, the set of program instructions comprising:
>
> instructions for creating a bid manager agent for issuing a call for bids for usage of said resources, receiving said bids and selecting a best bid from among said bids, wherein each of said bids defines a predetermined context;
>
> instructions for creating a plurality of bidder agents for issuing said bids according to predetermined bidding policies in response to said call for bids, wherein one of said bidder agents issues said best bid and provides said resources upon selection of said best bid by said bid manager agent; and
>
> instructions for creating a plurality of resource_adapters for providing a uniform interface to access application program interfaces of said resources, one of said resource adapters being a caching adapter for maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents, and for receiving from said bid manager agent said call for bids and issuing said cached bids to said bid manager agent instead of requiring said predetermined bidder agents to issue said bids, and a no-caching adapter for receiving from said bid manager agent said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder ~~gents~~ agents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager agent.

An Advisory Action issued August 22, 2007, indicating that the request for reconsideration was considered but did not place the application in condition for allowance. Further, for purposes of appeal, the proposed amendments were not entered on the basis that they raise new issues that would require further consideration and/or search. No further claim amendments were submitted with the Notice of Appeal.

It should be noted prior to the filing of the Amendment After Final, a telephone interview was conducted on August 16, 2007 between the Examiner and applicants' counsel. At that time, the Examiner's rejection of claim 1 based on a lack of enablement

with regard to "instructions for creating" was discussed. It was applicants' understanding that the rejection primarily concerned the apparent lack of consistency between the preamble and the elements of the claim. Accordingly, applicants amended claim 1 to clarify, in the preamble, that the set of program instructions "create agents and adapters for optimizing a bidding process for resources." This amendment was consistent with the language of the claim elements. Under these circumstances, the Examiner's refusal to enter the proposed amendments was quite surprising, since they related simply to the preamble and a typographical error and did not raise new issues. As such, the amendments should have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

A concise explanation of the subject matter defined in each of the independent claims involved in the appeal (1, 8 and 17) is provided.

The claims do not stand or fall together. Each claim is to be considered by the Board in view of the arguments and comments submitted herein.

The subject matter of Claim 1 is directed to storage medium storing a set of program instructions executable on a data processing device and usable to optimize a bidding process for resources. The set of program instructions comprises instructions for creating a bid manager agent for issuing a call for bids for usage of said resources, receiving said bids and selecting a best bid from among said bids, wherein each of said bids defines a predetermined context (pg. 4, lines 22-24); instructions for creating a plurality of bidder agents for issuing said bids according to predetermined bidding policies in response to said call for bids, wherein one of said bidder agents issues said best bid and provides said resources upon selection of said best bid by said bid manager (pg. 4, lines 25-27); and instructions for creating a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources (pg. 5, lines 30-31), one of said resource adapters being a caching adapter for maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents, and for receiving from said bid manager said call for bids and issuing said cached bids to said bid manager instead of requiring said predetermined bidder agents to issue said bids (pg. 6, lines 12-24 and FIGS. 2A, 2B and 3), and a no-caching adapter for receiving from said bid manager said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder agents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager (pg. 6, lines 12-24 and FIGS. 2A, 2B and 3).

The subject matter of Claim 8 is directed to an optimized method of acquiring bids from a plurality of bidder agents for resources (see pg. 9, line 8 to pg. 11, line 15, and FIGS. 11 and 12). The method comprises the steps of: issuing a request for bids for usage of said resources, wherein each of said bids defines a predetermined context; accessing a cache of stored bids and related contexts to determine whether said cache

contains bids defining said predetermined context; issuing a call for bids to said bidder agents in connection with which no bids defining said predetermined context are stored in said cache, in response to which said bidder agents return bids to said bid manager and said bids are stored in said cache along with said predetermined context; and retrieving from said cache said bids defining said predetermined context previously stored by said bidder agents.

The subject matter of claim 17 is directed to an apparatus for optimizing a bidding process for resources (see pg. 3, line 30 to pg. 6, line 24, and FIGS. 1-3). The apparatus comprises: a bid manager agent (pg. 4, lines 22-24, and FIGS. 1-3) comprising means for issuing a call to bidder agents for bids for usage of said resources (pg. 4, lines 22-24, and FIG. 1), means for receiving said bids and means for selecting a best bid from among said bids (pg. 4, lines 22-24, and FIG. 1), wherein each of said bids defines a predetermined context 1 or 2 (pg. 4, lines 3-13 and FIGS. 7-10); and a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources (pg. 5, lines 30-31, and FIGS. 2, 2A, and 2B), one of said resource adapters being a caching adapter (pg. 6, lines 12-21, and FIG. 2B) comprising means for maintaining cached bids for predetermined contexts 1 or 2 from predetermined ones of said bidder agents (pg. 4, lines 29-31, FIG. 2B), receiving from said bid manager said call for bids and issuing said cached bids to said bid manager instead of requiring said predetermined bidder agents to issue said bids (pg. 6, lines 12-24, and FIGS. 2A, 2B, 3, and 9), and a no-caching adapter (pg. 6, lines 12-21, and FIG. 2B) comprising means for receiving from said bid manager said call for bids (pg. 6, lines 12-21, and FIG. 2B), re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder agents (pg. 6, lines 12-21, and FIG. 2B), receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager (pg. 6, lines 12-21, and FIGS. 2A, 2B and 3).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following grounds of rejection are presented for review:

A) Whether claims 1-7, 17 satisfy the requirements of 35 U.S.C. 112, first paragraph.

B) Whether claims 1-7 satisfy the requirements of 35 U.S.C. 112, second paragraph.

C) Whether claims 1-7 are unpatentable as having been obvious under 35 U.S.C. §103(a) over Johnson (U.S. Patent No. 6,005,925) in view of Yee (U.S. Patent No. 6,738,975) in view of Baindur, et al. (U.S. Pat. No. 6,073,176) and further in view of Kou (U.S. Patent No. 6,363,365).

D) Whether claims 8-16 are unpatentable as having been obvious under 35 U.S.C. §103(a) over Johnson (U.S. Patent No. 6,005,925) in view of Yee (U.S. Patent No. 6,738,975) in view of Baindur, et al. (U.S. Pat. No. 6,073,176) and further in view of Kou (U.S. Patent No. 6,363,365).

E) Whether claim 17 is unpatentable as having been obvious under 35 U.S.C. §103(a) over Johnson (U.S. Patent No. 6,005,925) in view of Yee (U.S. Patent No. 6,738,975) in view of Baindur, et al. (U.S. Pat. No. 6,073,176) and further in view of Kou (U.S. Patent No. 6,363,365).

VII. ARGUMENT

A. Claims 1-7, 17 Satisfy The Requirements Of 35 U.S.C. 112, First Paragraph

In the Office Action mailed September 13, 2006, the Examiner rejected claims 1-7, 17 under 35 U.S.C. 112, first paragraph, asserting that they fail to comply with the enablement requirement. The Examiner stated that these claims contain subject matter not described in the specification in a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The Examiner's comments were specifically directed to the term "resource adaptors" as recited in the claims. A similar rejection had been previously addressed and seemingly overcome. Nevertheless, applicants continue to stand by their previous argument that a person skilled in the relevant art would understand the subject matter in question.

The Examiner conceded that "resource adapter" is a class used in object oriented programming of the invention, which provides for a uniform interface to access APIs of resources. Figure 2 serves to illustrate this relationship. Exactly why the Examiner is unsure about this is unclear. The description goes on to present an application scenario in Figure 6 wherein the Bidders decide on which kind of resource adapter they want the bid manager to use for the bidding protocol. One skilled in the art would readily understand such an application scenario. As such, it is believed that any further definition would only serve to unduly narrow the scope for which patent protection is sought.

In the Final Office Action mailed March 21, 2007, the Examiner continued to reject claims 1-7, 17 for containing subject matter that is not described in the specification in such a way as to enable one skilled in the art to make and/or use the invention. Specifically, the Examiner opined that the recited "resource adapters" are not well defined and that "instructions for creating" are nowhere described in the specification.

In response to applicants' previous arguments on this point, the Examiner stated that "the applicant has not provided support or explanation for resource adapters sufficient to allow an understanding of how to use the invention." Without wishing to concede any relevance of Yee et al. (US 6,738,975) as a reference, the Examiner's

assertion that it is not clear how a resource adapter provides a uniform interface to access application programming interfaces of resources appears to be inconsistent with her reliance on Yee et al. to establish that it would have an obvious to modify Johnson (US 6,005,925) to include a resource adapter as taught by Yee et al.

In any event, it is respectfully submitted that resource adapters (including their structure and use) were notoriously well-known as of the filing date of this application. As but one example of the numerous well-known technical specifications relating to resource adapters, applicants have attached a copy of a draft specification for Java Transaction Service that refers extensively to the operation and design of resource adapters and resource managers and that is completely consistent with Figures 2A and 2B (see, for example, the definition at the bottom of page 4 of the attached document). Many other examples are available that make it clear resource adapters would be well understood by a person of ordinary skill in the art at the time of the invention (*see, e.g.*, U.S. Patent Nos. 5,165,031 and 5,613,060).

The Examiner has also argued that a description of the "instructions for creating" could not be found in the specification to enable one skilled in the art to make and/or use the invention. In this regard, it is noted that the present invention is directed to software. See, for example, FIGS. 4 and 5, which show pseudo-code for implementing aspects of the invention, and FIG. 11, which is a flowchart showing the caching process. Under the circumstances, it is to be understood that the software described in the specification, including a set of program instructions for creating the various actors and use cases, may be stored on a storage medium.

Accordingly, it is respectfully submitted that the Examiner's rejection based on lack of enablement in connection with the recited "resource adapters" and "instructions for creating" has been traversed.

B. Claims 1-7 Satisfy The Requirements Of 35 U.S.C. 112, Second Paragraph

The Examiner has also rejected claim under 35 U.S.C. 112, second paragraph, asserting the claim is indefinite. Independent claim 1 relates to a computer-readable storage medium encoded with a data structure defining structural and functional interrelationships between the data structure and the computer software and hardware

components which permit the data structure's functionality to be realized and is thus statutory. See, for example, MPEP 2106.01. The Examiner, however, opines that the claim is unclear because it does not achieve the result disclosed in the preamble, that is, the claim recites "creating" various items but does not actually say that they are run or executed. As noted earlier, a description of the "instructions for creating" can be found in the specification. See, for example, FIGS. 4 and 5, which show pseudo-code for implementing aspects of the invention, and FIG. 11, which is a flowchart showing the caching process. Under the circumstances, it is to be understood that the software described in the specification, including a set of program instructions for creating the various actors and use cases, may be stored on a storage medium. Thus, claim 1 is clear.

Accordingly, applicants submit that claims 1-7 satisfy the requirements of 35 U.S.C. 112, second paragraph, and are thus allowable.

C. Claims 1-7 Would Not Have Been Obvious Over Johnson in view of Yee in view of Baindur, et al. and in further view of Kou.

The Examiner rejected claims 1-7 under 35 U.S.C. 103(a) as being unpatentable over Johnson (U.S. Patent No. 6,005,925) in view of Yee (U.S. Patent No. 6,738,975) in view of Baindur (U.S. Patent No. 6,073,176) in further view of Kou (U.S. Patent No. 6,363,365). As presented in *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed.Cir. 1990), the mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination.

In the present application, claim 1, as amended, recites, inter alia, "instructions for creating a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources, one of said resource adapters being a caching adapter for maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents, and for receiving from said bid manager agent said call for bids and issuing said cached bids to said bid manager agent instead of requiring said predetermined bidder agents to issue said bids, and a no-caching adapter for receiving from said bid manager agent said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder gents,

receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager agent."

In assembling the rejection of claims 1-7, the general concept of a cache for storing bids is not presented until the final reference (Kou), after two previous prior art combinations. Applicants submit that the "desirability of the combination" is just not present in the prior art. Further, it seems highly unlikely that such a combination would have been obvious without the assistance of hindsight. As stated in *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971), "so long as [the hindsight reconstruction] takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper."

Applicants respectfully submit that the reconstruction presented by the Examiner is improper as the use of a Caching Adapter within a multi-agent caching system "for maintaining cached bids for predetermined contexts" falls squarely into the realm of "knowledge gleaned only from the applicants' disclosure." One would not have been able to arrive at the present reconstruction presented in the Action without such knowledge from the disclosure. This is a classic example of impermissible hindsight.

In the Final Office Action, the Examiner did not elaborate on the obviousness rejection set forth in the previous Office Action, other than by responding with a divergent opinion on the correct standard for obviousness. Specifically, in her "Response to Arguments," the Examiner acknowledged that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, and concluded that in the present case, "the cited references pertain to analogous art i.e. e-commerce, electronic auctions." That said, the mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification. *See In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed.Cir. 1984). Further, both the suggestion and the expectation of success must be founded in the prior art, not in applicant's disclosure. *See In re Dow Chemical Co. v. American Cyanamid Co.*, 837 F.2d 469, 5 USPQ2d

1529 (Fed.Cir. 1988). And as noted in *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. __, 127 S.Ct. 1727 (2007), rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. That said, it is also true that the prior art references must teach or suggest all of the limitations of the claims. And as noted below, all of the elements of the claims are not taught or suggested by the cited references.

Turning from the debate regarding the legal standard for obviousness to the facts of this application, Johnson is cited for disclosing instructions for creating "a bid manager agent for issuing a call for bids for usage of said resources, receiving said bids and selecting a best bid from among said bids, wherein each of said bids defines a predetermined context." Johnson is further cited for creating "a plurality of bidder agents for issuing said bids according to predetermined bidding policies in response to said call for bids."

The Examiner conceded that Johnson does not specifically disclose "a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources." Yee et al. is cited for teaching "an adapter in order to transform the data from one application so it can be used by other application [sic]." Examiner's argument is confusing since the claim limitation missing from Johnson is "a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources," not "...transform the data from one application so it can be used by other application."

In any event, the Examiner conceded that Johnson and Yee et al. do not specifically disclose "issuing said cached bids to said bid manager agent instead of requiring said predetermined bid agents to issue said bid, and a no-caching adapter for receiving from said bid manager agent said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder agents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager agent."

The Examiner nonetheless concluded that it would have been obvious "to modify Johnson and Yee to include maintaining a default bid in memory to be used if desired

by the bidder in order to provide the bidder with various bidding options according to the bidder's capacity to process the event efficiently" (emphasis added). The Examiner's conclusion is confusing since claim 1 does not recite "maintaining a default bid in memory," but rather "maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents." The term "context" is defined in the specification at page 4, lines 3-4 as "the set of values that the Bidders need to know in order to calculate their bids accordingly." Thus, if the context of a call is identical to one that has occurred previously, the bid manager consults the cached bids rather than sending a call for new bids to bidders that have previously submitted bids for that context (*i.e.*, the "predetermined ones of said bidder agents"). There is absolutely no teaching or suggestion in any of the cited references of applicant's recited "maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents."

The Examiner has also conceded that Johnson, Yee et al. and Baindur do not disclose using a cache to store bids and cites Kou for teaching a bid cache. The bid cache of Kou is used to store bids from multiple bidders until the closing day of the tender, whereupon the bid requester opens all bid proposals and selects the successful tender. As discussed above in connection with Johnson, Yee et al. and Baindur, there is no teaching or suggestion in Kou of applicant's recited "maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents."

As applicants consider all of the Examiner's arguments traversed, the rejection of claims 1-7 under 35 U.S.C. §103(a) must be reversed.

D. Claims 8-16 Would Not Have Been Obvious Over Johnson in view of Yee in view of Baindur, et al. and in further view of Kou.

Independent claim 8 recites an optimized method of acquiring bids from a plurality of bidder agents for resources. More particularly, claim 8 includes the steps of:

> issuing a request from a bid manager agent for bids for usage of said resources, wherein each of said bids defines a predetermined context;
>
> accessing a cache of stored bids and related contexts to determine whether said cache contains bids defining said predetermined context;

> issuing a call for bids to said bidder agents in connection with which no bids defining said predetermined context are stored in said cache, in response to which said bidder agents return bids to said bid manager agent and said bids are stored in said cache along with said predetermined context; and
>
> retrieving from said cache said bids defining said predetermined context previously stored by said bidder agents.

The Examiner's rejection of claim 8 relies upon the reasons cited by the Examiner in rejecting claim 1:

> Re claims 8-16: Further a method would have been necessary to perform the method of previously rejected claims 1-7 and are therefore rejected using the same art and rationale.

The cited references, however, fail to teach or suggest the features of claim 8 as noted above.

In assembling the rejection of claims 1-7, the general concept of a cache for storing bids is not presented until the final reference (Kou), after two previous prior art combinations. Applicants submit that the "desirability of the combination" is just not present in the prior art. Further, it seems highly unlikely that such a combination would have been obvious without the assistance of hindsight. As stated in *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971), "so long as [the hindsight reconstruction] takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper."

Applicants respectfully submit that the reconstruction presented by the Examiner is improper as the use of a cache within a multi-agent caching system for maintaining cached bids for predetermined contexts falls squarely into the realm of "knowledge gleaned only from the applicants' disclosure." One would not have been able to arrive at the present reconstruction presented in the Action without such knowledge from the disclosure. This is a classic example of impermissible hindsight.

In the Final Office Action, the Examiner did not elaborate on the obviousness rejection set forth in the previous Office Action, other than by responding with a divergent opinion on the correct standard for obviousness. Specifically, in her "Response to Arguments," the Examiner acknowledged that obviousness can only be

established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, and concluded that in the present case, "the cited references pertain to analogous art i.e. e-commerce, electronic auctions." That said, the mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification. *See In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed.Cir. 1984). Further, both the suggestion and the expectation of success must be founded in the prior art, not in applicant's disclosure. *See In re Dow Chemical Co. v. American Cyanamid Co.*, 837 F.2d 469, 5 USPQ2d 1529 (Fed.Cir. 1988). And as noted in *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. __, 127 S.Ct. 1727 (2007), rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. That said, it is also true that the prior art references must teach or suggest all of the limitations of the claims. And as noted below, all of the elements of the claims are not taught or suggested by the cited references.

Turning from the debate regarding the legal standard for obviousness to the facts of this application, Johnson is presumably cited for disclosing the steps of "issuing a request from a bid manager agent for bids for usage of said resources, wherein each of said bids defines a predetermined context" and "accessing a cache of stored bids and related contexts to determine whether said cache contains bids defining said predetermined context."

The Examiner conceded that Johnson and Yee et al. do not specifically disclose issuing said cached bids to said bid manager agent instead of requiring said predetermined bid agents to issue said bid, and a no-caching adapter for receiving from said bid manager agent said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder agents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager agent.

The Examiner nonetheless concluded that it would have been obvious "to modify Johnson and Yee to include maintaining a default bid in memory to be used if desired by the bidder in order to provide the bidder with various bidding options according to the bidder's capacity to process the event efficiently" (emphasis added). The Examiner's conclusion is confusing since claim 8 does not recite "maintaining a default bid in memory," but rather maintaining a cache of stored bids and related contexts to determine whether said cache contains bids defining said predetermined context. The term "context" is defined in the specification at page 4, lines 3-4 as "the set of values that the Bidders need to know in order to calculate their bids accordingly." Thus, if the context of a call is identical to one that has occurred previously, the bid manager consults the cached bids rather than sending a call for new bids to bidders that have previously submitted bids for that context (*i.e.*, the predetermined ones of said bidder agents). There is absolutely no teaching or suggestion in any of the cited references of the features of claim 8.

The Examiner has also conceded that Johnson, Yee et al. and Baindur do not disclose using a cache to store bids and cites Kou for teaching a bid cache. The bid cache of Kou is used to store bids from multiple bidders until the closing day of the tender, whereupon the bid requester opens all bid proposals and selects the successful tender. As discussed above in connection with Johnson, Yee et al. and Baindur, there is no teaching or suggestion in Kou of maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents.

As applicants consider all of the Examiner's arguments traversed, the rejection of claims 8-16 under 35 U.S.C. §103(a) must be reversed.

E. Claim 17 Would Not Have Been Obvious Over Johnson in view of Yee in view of Baindur, et al. and in further view of Kou.

Independent claim 17 recites an apparatus for optimizing a bidding process for resources. More particularly, claim 17 includes "a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources, one of said resource adapters being a caching adapter comprising means for maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents, receiving from said bid manager said call for bids and issuing said

cached bids to said bid manager instead of requiring said predetermined bidder agents to issue said bids, and a no-caching adapter comprising means for receiving from said bid manager said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder agents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager."

The Examiner's rejection of claim 17 relies upon many of the reasons that have been cited by the Examiner in rejecting claim 1:

> Re claim 17: Further an apparatus would have been necessary to perform the method of previously rejected claims 1-7 and are therefore rejected using the same art and rationale.

In assembling the rejection of claims 1-7, the general concept of a cache for storing bids is not presented until the final reference (Kou), after two previous prior art combinations. Applicants submit that the "desirability of the combination" is just not present in the prior art. Further, it seems highly unlikely that such a combination would have been obvious without the assistance of hindsight. As stated in *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971), "so long as [the hindsight reconstruction] takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper."

Applicants respectfully submit that the reconstruction presented by the Examiner is improper as the use of a Caching Adapter within a multi-agent caching system "for maintaining cached bids for predetermined contexts" falls squarely into the realm of "knowledge gleaned only from the applicants' disclosure." One would not have been able to arrive at the present reconstruction presented in the Action without such knowledge from the disclosure. This is a classic example of impermissible hindsight.

In the Final Office Action, the Examiner did not elaborate on the obviousness rejection set forth in the previous Office Action, other than by responding with a divergent opinion on the correct standard for obviousness. Specifically, in her "Response to Arguments," the Examiner acknowledged that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so

found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, and concluded that in the present case, "the cited references pertain to analogous art i.e. e-commerce, electronic auctions." That said, the mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification. *See In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed.Cir. 1984). Further, both the suggestion and the expectation of success must be founded in the prior art, not in applicant's disclosure. *See In re Dow Chemical Co. v. American Cyanamid Co.*, 837 F.2d 469, 5 USPQ2d 1529 (Fed.Cir. 1988). And as noted in *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. __, 127 S.Ct. 1727 (2007), rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. That said, it is also true that the prior art references must teach or suggest all of the limitations of the claims. And as noted below, all of the elements of the claims are not taught or suggested by the cited references.

Turning from the debate regarding the legal standard for obviousness to the facts of this application, Johnson is cited for disclosing a bid manager agent for issuing a call for bids for usage of said resources, receiving said bids and selecting a best bid from among said bids, wherein each of said bids defines a predetermined context.

The Examiner conceded that Johnson does not specifically disclose "a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources." Yee et al. is cited for teaching "an adapter in order to transform the data from one application so it can be used by other application [sic]." Examiner's argument is confusing since the claim limitation missing from Johnson is "a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources," not "...transform the data from one application so it can be used by other application."

In any event, the Examiner conceded that Johnson and Yee et al. do not specifically disclose "issuing said cached bids to said bid manager agent instead of requiring said predetermined bid agents to issue said bid, and a no-caching adapter for receiving from said bid manager agent said call for bids, re-issuing said call for bids to

ones of said bidder agents other than said predetermined bidder agents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager agent,"

The Examiner nonetheless concluded that it would have been obvious "to modify Johnson and Yee to include maintaining a default bid in memory to be used if desired by the bidder in order to provide the bidder with various bidding options according to the bidder's capacity to process the event efficiently" (emphasis added). The Examiner's conclusion is confusing since claim 17 does not recite "maintaining a default bid in memory," but rather "maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents" (emphasis added). The term "context" is defined in the specification at page 4, lines 3-4 as "the set of values that the Bidders need to know in order to calculate their bids accordingly." Thus, if the context of a call is identical to one that has occurred previously, the bid manager consults the cached bids rather than sending a call for new bids to bidders that have previously submitted bids for that context. (*i.e.*, the "predetermined ones of said bidder agents"). There is absolutely no teaching or suggestion in any of the cited references of applicant's recited "maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents."

The Examiner has also conceded that Johnson, Yee et al. and Baindur do not disclose using a cache to store bids and cites Kou for teaching a bid cache. The bid cache of Kou is used to store bids from multiple bidders until the closing day of the tender, whereupon the bid requester opens all bid proposals and selects the successful tender. As discussed above in connection with Johnson, Yee et al. and Baindur, there is no teaching or suggestion in Kou of applicant's recited "maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents."

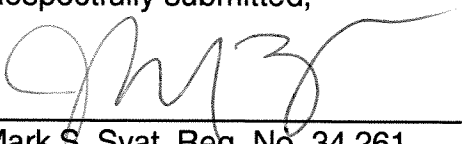
As applicants consider all of the Examiner's arguments traversed, the rejection of claim 17 under 35 U.S.C. §103(a) must be reversed.

VIII. CONCLUSION

For the reasons discussed above, it is respectfully submitted that the rejections are in error and that claims 1-17 are in condition for allowance. For all of the above

reasons, Appellants respectfully request this Honorable Board to reverse the rejections of claims 1-17.

Respectfully submitted,



Mark S. Svat, Reg. No. 34,261
John S. Zanghi, Reg. No. 48,843

MSS:JSZ/ec

FAY, SHARPE, FAGAN, MINNICH & MCKEE, LLP
1100 Superior Avenue – Seventh Floor
Cleveland, Ohio 44114-2579
Telephone: (216) 861-5582

Filed: 11/20/07

CLAIMS APPENDIX

CLAIMS INVOLVED IN THE APPEAL:

1. (Previously Presented) A storage medium storing a set of program instructions executable on a data processing device and usable to optimize a bidding process for resources, the set of program instructions comprising:

instructions for creating a bid manager agent for issuing a call for bids for usage of said resources, receiving said bids and selecting a best bid from among said bids, wherein each of said bids defines a predetermined context;

instructions for creating a plurality of bidder agents for issuing said bids according to predetermined bidding policies in response to said call for bids, wherein one of said bidder agents issues said best bid and provides said resources upon selection of said best bid by said bid manager agent; and

instructions for creating a plurality of resource_adapters for providing a uniform interface to access application program interfaces of said resources, one of said resource adapters being a caching adapter for maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents, and for receiving from said bid manager agent said call for bids and issuing said cached bids to said bid manager agent instead of requiring said predetermined bidder agents to issue said bids, and a no-caching adapter for receiving from said bid manager agent said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder gents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager agent.

2. (Previously Presented) The storage medium of claim 1, wherein said program instructions further comprise instructions for updating said cached bids in response to new contexts of said bids.

3. (Previously Presented) The storage medium of claim 1, wherein said program instructions further comprise instructions for selecting said best bid by sorting said resource adapters according to decreasing values of said bids and selecting a first available one of said bidder agents according to said resource adapters as sorted.

4. (Previously Presented) The storage medium of claim 1, wherein each said context is defined by a discrete parameter value.

5. (Previously Presented) The storage medium of claim 1, wherein said program instructions further comprise instructions for sending a notification message to said bid manager agent in the event of any changes to its bidding policies, in response to which said bid manager agent updates said caching adapter.

6. (Previously Presented) The storage medium of claim 5, wherein said program instructions further comprise instructions for storing said bidding policies via said caching adapter as entries in a table and updating individual ones of said cached bids to reflect said changes in said bidding policies.

7. (Previously Presented) The storage medium of claim 5, wherein said program instructions further comprise instructions for are storing said bidding policies via said caching adapter as general rules and clearing all of said cached bids.

8. (Previously Presented) An optimized method of acquiring bids from a plurality of bidder agents for resources, comprising the steps of:

issuing a request from a bid manager agent for bids for usage of said resources, wherein each of said bids defines a predetermined context;

accessing a cache of stored bids and related contexts to determine whether said cache contains bids defining said predetermined context;

issuing a call for bids to said bidder agents in connection with which no bids defining said predetermined context are stored in said cache, in response to which said bidder agents return bids to said bid manager agent and said bids are stored in said cache along with said predetermined context; and

retrieving from said cache said bids defining said predetermined context previously stored by said bidder agents.

9. (Original) The optimized method of claim 8, further comprising the step of updating said stored bids in response to new contexts of said bids.

10. (Previously Presented) The optimized method of claim 8, further comprising the step of selecting a best bid by sorting said bids according to decreasing values of said bids and selecting a first available one of said bidder agents according to said sorting.

11. (Original) The optimized method of claim 8, wherein each said context is defined by a discrete parameter value.

12. (Previously Presented) The optimized method of claim 8, further comprising the step of sending a notification message to said bid manager agent in the event of any changes to its bidding policies, in response to which said bid manager agent updates said cache.

13. (Original) The optimized method of claim 12, further comprising the step of storing said bidding policies as entries in a table.

14. (Original) The optimized method of claim 13, further comprising the step of updating individual ones of said cached bids for updating said cache to reflect said changes in said bidding policies.

15. (Original) The optimized method of claim 12, further comprising the step of storing said bidding policies as general rules.

16. (Original) The optimized method of claim 15, further comprising the step of clearing all of said cached bids for updating said cache to reflect said changes in said bidding policies.

17. (Previously Presented) An apparatus for optimizing a bidding process for resources, the apparatus comprising:

a bid manager agent comprising means for issuing a call to bidder agents for bids for usage of said resources, means for receiving said bids and means for selecting a best bid from among said bids, wherein each of said bids defines a predetermined context;

a plurality of resource adapters for providing a uniform interface to access application program interfaces of said resources, one of said resource adapters being a caching adapter comprising means for maintaining cached bids for predetermined contexts from predetermined ones of said bidder agents, receiving from said bid manager agent said call for bids and issuing said cached bids to said bid manager agent instead of requiring said predetermined bidder agents to issue said bids, and a no-caching adapter comprising means for receiving from said bid manager agent said call for bids, re-issuing said call for bids to ones of said bidder agents other than said predetermined bidder agents, receiving said bids from said ones of said bidder agents other than said predetermined bidder agents and sending said bids to said bid manager agent.

EVIDENCE

Java Transaction Service (JTS) document attached.

RELATED PROCEEDINGS

Not applicable.

Sun Microsystems Inc.

Java Transaction Service (JTS)

This is a draft specification for Java Transaction Service (JTS). JTS specifies the implementation of a transaction manager which supports the JTA specification [1] at the high-level and implements the Java mapping of the OMG Object Transaction Service (OTS) 1.1 Specification at the low-level.

JTS uses the CORBA OTS interfaces for interoperability and portability, which defines a standard mechanism for any implementation that utilizes IIOP (Internet InterORB Protocol) to generate and propagate transaction context between JTS Transaction Managers.

Please send technical comments on this specification to:

jts-spec@eng.sun.com

Version 0.95

Susan Cheung
March 01, 1999

Table of Contents

1 Introduction

This is the Java Transaction Service (JTS) Specification. JTS specifies the implementation of a transaction manager which supports the JTA specification [1] at the high-level and implements the Java mapping of the OMG Object Transaction Service (OTS) 1.1 Specification at the low-level.

JTS uses the CORBA OTS interfaces for interoperability and portability (that is, CosTransactions and CosTSPortability). These interfaces define a standard mechanism for any implementation that utilizes IIOP (Internet InterORB Protocol) to generate and propagate transaction context between JTS Transaction Managers. Note, this also permits the use of other API over the IIOP transport mechanism to be used; for example, RMI over IIOP is allowed.

1.1 Background

Distributed transaction services in Enterprise Java middleware involve five players: the transaction manager, the application server, the resource manager, the application program, and the communication resource manager. Each player contributes to the distributed transaction processing system by implementing different sets of transaction APIs and functionalities.

- A transaction manager provides the services and management functions required to support transaction demarcation, transactional resource management, synchronization, and transaction context propagation.
- An application server (or TP monitor) provides the infrastructure required to support the application run-time environment which includes transaction state management. An example of such an application server is an EJB [5] server.
- A resource manager (through a resource adapter[1]) provides the application access to resources. The resource manager implements a transaction resource interface that is used by the transaction manager to communicate transaction association, transaction completion, and recovery work. An example of such a resource manager is a relational database server.
- A component-based transactional application that operates in a modern application server environment relies on the application server to provide transaction management support through declarative transaction attribute settings—for example, an application developed using the industry standard Enterprise JavaBeans (EJB) component architecture. In addition, other stand-

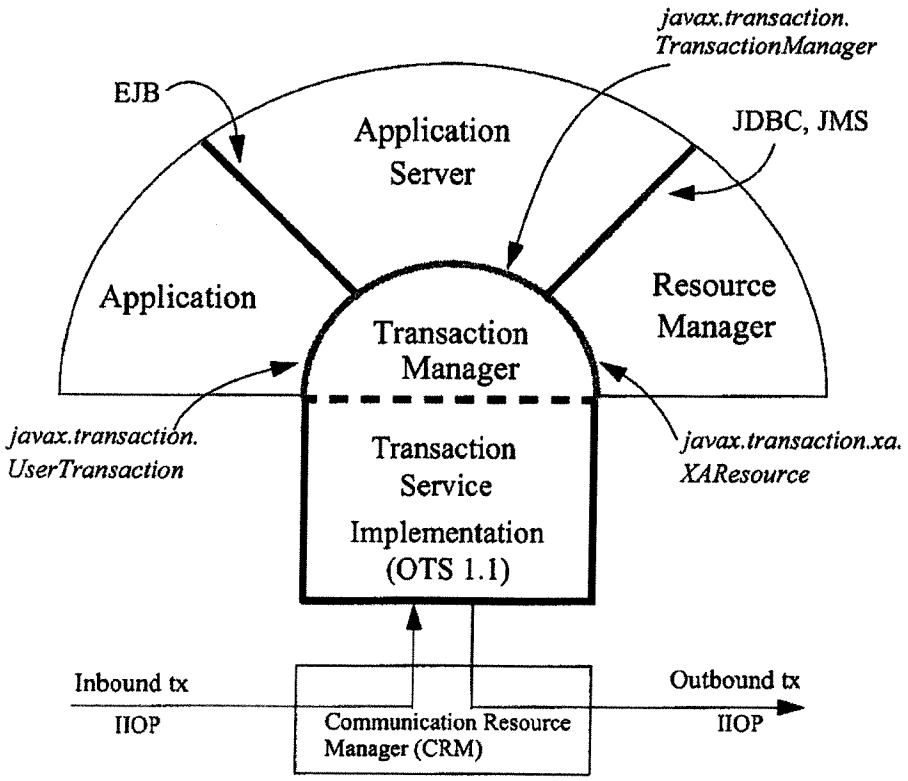
1.A Resource Adapter is a system level software library that is used by an application server or client to connect to a Resource Manager. A Resource Adapter is typically specific to a Resource Manager. It is available as a library and is used within the address space of the client using it. Examples of Resource adapters are: JDBC driver to connect to relational databases, ODMG driver to connect to an object database, JRFC library to connect to SAP R/3 system. A resource adapter may provide additional services besides the connection API.

alone Java client programs may wish to control their transaction boundaries using a high-level interface provided by the application server or the transaction manager.

- A communication resource manager (CRM) supports transaction context propagation and access to the transaction service for incoming and outgoing requests. The JTS document does not specify requirements pertaining to communication. We assume the CRM is present to support transaction propagation as defined in the CORBA OTS and GIOP specifications.

From the transaction manager's perspective, the actual implementation of the transaction services does not need to be exposed; only high-level interfaces need to be defined to allow transaction demarcation, resource enlistment, synchronization, and recovery process to be driven by the users of the transaction services.

The diagram below shows the high-level API exposed from the transaction manager that implements the JTS specification. The dotted-line in the Transaction Manager box illustrates the private interface within the TM to allow the JTA support module to interact with the low-level OTS implementation. Section 2 specifies the Transaction Manager external functionality. Section 3 specifies the Transaction Manager implementation requirements and considerations.



1.2 Target Audience

This document is intended for implementors of Transaction Managers and application servers written in the Java[tm] programming language.

2 Transaction Manager Functionality

This section describes the transaction manager functionality through support of the Java Transaction API (JTA). The implementation of the Java mapping of OMG OTS 1.1 interfaces are not exposed to the clients of the Transaction Manager. The clients of the Transaction Manager are those who use the JTA interfaces to access the Transaction Manager functionality.

The Transaction Manager provides the following services:

- Provides applications and application servers the ability to control the scope and duration of a transaction.
- Allows multiple application components to perform work that is part of a single, atomic transaction.
- Provides the ability to associate global transactions with work performed by transactional resources.
- Coordinates the completion of global transactions across multiple resource managers.
- Supports transaction synchronization.
- Provides the ability to interoperate with other Transaction Manager implementations using the CORBA ORB/TS standard interfaces. (This is transparent to clients of the Transaction Manager.)

2.1 Transaction Model

The Transaction Manager is required to support distributed flat transactions. A flat transaction cannot have a child transaction. Flat transactions are also known as top-level transactions in OTS terminology. A Transaction is started by issuing a request to begin a transaction.

Support for nested transactions is not required.

2.2 Transaction Context

The Transaction Manager maintains the association of a thread's transaction context with a transaction. A thread's transaction context is either *null* or refers to a specific global transaction. The Transaction Manager allows multiple threads to be associated with the same transaction concurrently, in the same JVM or in multiple JVMs.

Transaction context is implicitly transmitted by the implementation of the transaction service at the ORB and wire-protocol level. The transaction context propagation is performed transparent to the Transaction Manager clients (application and application server).

2.3 Transaction Termination

A transaction is terminated by issuing a request to commit or rollback the transaction. Typically, a transaction is terminated by the client originating the transaction. In the

EJB component model environment, the Transaction Manager must allow transactions to be terminated by any thread within the same JVM of the transaction originator.

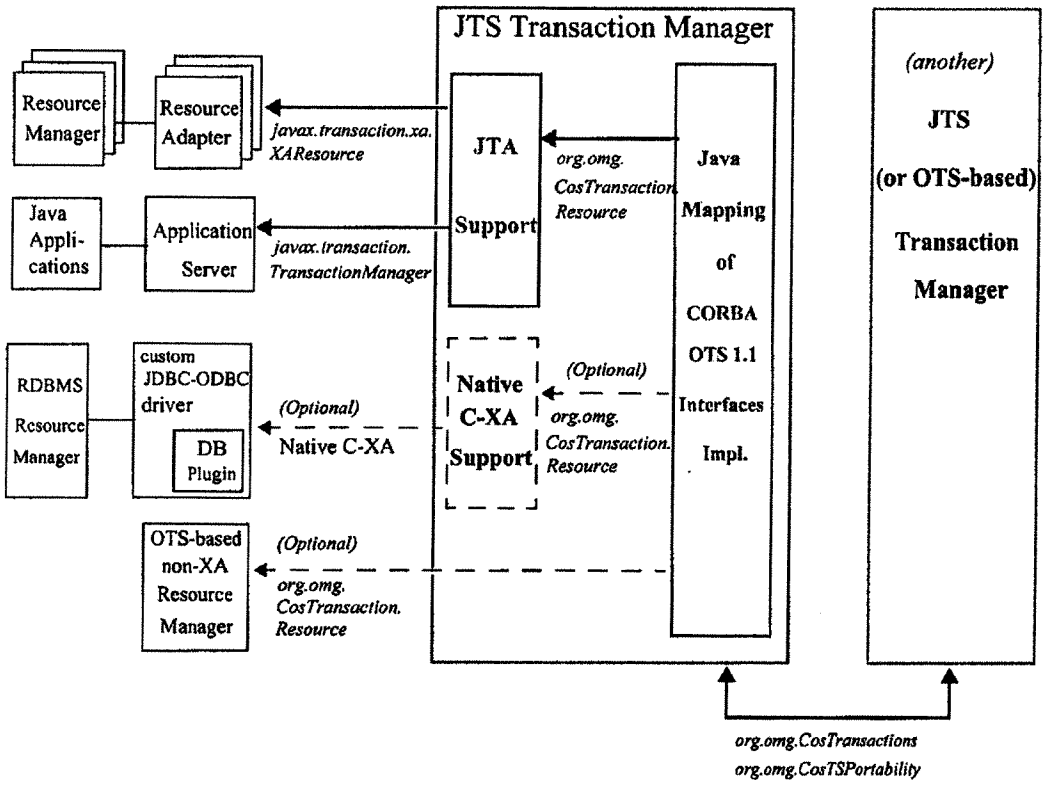
Application components that rely on an application server to manage their transaction states are not allowed to terminate transactions. An application server can force the transaction to be rolled back after the application encounters an unexpected error condition in the form of a Java exception. The Transaction Manager is not required to monitor the failures of the resource managers participating in the transaction.

2.4 Transaction Integrity

The Transaction Manager is required to guarantee data integrity equivalent to that provided by the interfaces which support the X/Open DTP transaction model. The Transaction Manager must guarantee the checked transaction behavior—a transaction cannot be committed until all computations acting on behalf of the transaction have completed.

3 Transaction Manager Implementation

This section describes the implementation choices from a Transaction Manager implementor's view. As shown in the diagram below, the Transaction Manager must implement the JTA interfaces to support the application server and the resource managers. Support for JDBC 1.0 driver and non-JTA aware resource managers is optional. Support for direct CORBA clients, such as recoverable servers and transactional objects, is also optional.



3.1 Support For JTA

The Transaction Manager provides complete support of the Java Transaction API (JTA) Specification [1].

3.1.1 Transaction Demarcation

The Transaction Manager implements the following JTA interfaces to allow application servers and stand-alone Java client applications to control transaction boundary demarcation and perform transaction operations.

- *javax.transaction.TransactionManager*
- *javax.transaction.Transaction*
- *javax.transaction.UserTransaction*

3.1.2 Transaction Synchronization

The Transaction Manager supports transaction synchronization by allowing Synchronization callback objects to be registered by the application server. The Transaction Manager invokes the Synchronization methods before and after transaction completion. Synchronization registration is available via the `javax.transaction.Transaction.registerSynchronization` method.

3.1.3 Transaction and Resource Association

The Transaction Manager supports transactional resource enlistment via the `enlistResource` and `delistResource` methods defined in the `javax.transaction.Transaction` interface.

The Transaction Manager associates resources with transactions and coordinates transaction completion using the `javax.transaction.xa.XAResource` interface as defined in JTA.

3.1.4 Transaction Recovery

The Transaction Manager uses the `recover` and `forget` methods in the `javax.transaction.xa.XAResource` interface to recover transactions that are in prepared or heuristically completed states.

3.2 Java Mapping of CORBA Object Transaction Service (OTS)

The Transaction Manager implements the Java Mapping of the CORBA Object Transaction Service 1.1 Specification [2]. In particular, the Transaction Manager implements the following Java packages: *org.omg.CosTransactions* and *org.omg.CosTSPortability*.

The Transaction Manager is not required to support nested transactions.

The Transaction Manager is not required to expose its OTS implementation to those users who are accessing the Transaction Manager through the `javax.transaction.TransactionManager` interface as defined in JTA.

3.3 Support for Pre-JTA Resource Managers

The Transaction Manager may optionally support pre-JTA resource managers. Specifically, the Transaction Manager may implement a native C-XA support module to provide transaction coordination using the native C-XA procedural interfaces as defined in the X/Open XA Specification [4].

As shown in the previous diagram, to support existing relational database servers that implement the C-XA procedural interface, the Transaction Manager implements a native C-XA support module which uses the `CosTransactions.Resource` interface

to interact with the transaction service module. External to the Transaction Manager, a custom JDBC driver will need to be implemented with a native-XA library built specific to each database server.

3.4 Support for CORBA Applications

The Transaction Manager may optionally support the following CORBA application entities as defined in the Object Transaction Specification: Transactional Client, Transactional Objects, Recoverable Objects, Transactional Servers, and Recoverable Servers. These application entities access the Transaction Manager using the interfaces defined in the *CosTransactions* module as specified in the OTS 1.1 Specification.

3.5 Transaction Managers Interoperability

The Transaction Manager is required to support distributed transactions that involve multiple resource managers in a single ORB execution environment.

If the Transaction Manager implementation supports inter-ORB interoperability, it must implement the implicit transaction context propagation that conforms to the `CosTransactions.PropagationContext` structure; this allows the Transaction Manager to support inter-ORB transaction context propagation as defined by the CORBA OTS 1.1 Specification.

To provide interaction between the ORB and the Transaction Manager, the Transaction Manager is required to

- Implement the `CosTSPortability` module's `Sender` and `Receiver` interfaces as callback objects to allow the ORB to notify the TM whenever a transaction request is sent or received by the ORB.
- Invoke the `TSIdentification` interface methods to pass the `Sender` and `Receiver` objects to the ORB, prior to handling the first transactional request.

How the ORB and the Transaction Manager locate each other's objects is discussed in section 3.6 below. The wire protocol message format for transmitting the transaction context is defined in the CORBA General Inter-ORB Protocol specification.

3.6 ORB Identification

The CORBA OTS 1.1 Specification does not define how the ORB and Transaction Manager identify each other. In order for different ORB instances and the Transaction Manager to interoperate and locate each other, JTS defines a simple `TransactionService` interface to facilitate the identification of the ORB to the Transaction Manager.

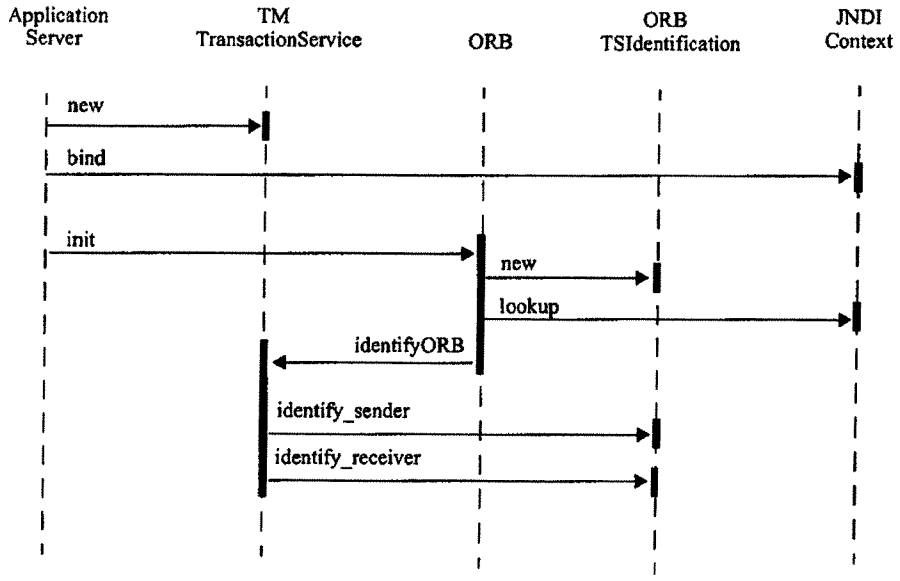
3.6.1 TransactionService Interface

The JTS Transaction Manager implements the `javax.jts.TransactionService` interface to allow an ORB to identify itself to the Transaction Manager.

The ORB calls the `TransactionService.identifyORB` method during its initialization procedure and prior to handling any user request.

Typically, the following operations occur:

1. The application server creates the `TransactionService` object
2. The application server binds the `TransactionService` object to the JNDI naming directory.
3. The application server initializes an ORB instance.
4. The ORB, during its initialization, creates a `TSIdentification` object and uses JNDI to lookup the `TransactionService` object reference.
5. The ORB then invokes the `TransactionService.identifyORB` method and supplies the following three parameters:
   - An ORB object that identifies the ORB instance.
   - A TSIdentification object implemented by the ORB.
   - A properties list for custom configuration information.
6. The Transaction Manager, while executing the `identifyORB` method, invokes the `TSIdentification.identify_sender` and `TSIdentification.identify_receiver` methods to pass the `Sender` and `Receiver` callback objects to the ORB.



Interface TransactionService

```
interface javax.jts.TransactionService {
   public void identifyORB(org.omg.CORBA.ORB orb,
      org.omg.TSIdentification tsi, Properties prop);
}
```

The `javax.jts.TransactionService` interface is implemented by the JTS Transaction Manager to allow the ORB to identify itself to the Transaction Manager and for the Transaction Manager to pass the Sender and Receiver callback objects to the ORB. The Sender and Receiver objects are used by the ORB to deliver the user request's transaction context to the Transaction Manager.

Methods

- **`identifyORB`**

```
public abstract void identifyORB(org.omg.CORBA.ORB orb,
                                 org.omg.CORBA.TSIdentification tsi,
                                 java.util.Properties prop);
)
```

The identifyORB method is called by the ORB as part of its initialization procedure.

Parameters:

`orb`
: The ORB instance

`tsi`
: The TSIdentification object for the TM to identify its Sender and Receiver callback objects.

`prop`
: The properties list for any customed information to the TM.

4 Related Documents

[1] Java Transaction API (JTA) Specification (*http://java.sun.com/products/jta*)

[2] OMG Object Transaction Service (*http://www.omg.org/corba/sectrans.html#trans*)

[3] ORB Portability Submission, OMG document orbos/97-04-14.

[4] X/Open CAE Specification – Distributed Transaction Processing: The XA Specification, X/Open Document No. XO/CAE/91/300 or ISBN 1 872630 24 3

[5] Enterprise JavaBeans™ Specification (*http://java.sun.com/products/ejb*)

[6] JDBC™ 2.0 Standard Extension API Specification (*http://java.sun.com/products/jdbc)*

[7] Java Message Service Specification (*http://java.sun.com/products/jms)*

Appendix A: Change History

A.1 Changes from 0.8 to 0.9

JTS revision 0.9 incorporated the following changes:

- Modified the diagram in Section 3 to include interoperability with another TM.
- Added section 3.6 to specify the `TransactionService` interface which allows the ORB and the TM to locate each other.

A.2 Changes from 0.9 to 0.95

- Added Copyright statement
- Minor editorial changes